Consciousness as a Computational Constraint Space

Y.Matsuda and ChatGPT5.2

January 10, 2026

Abstract

Consciousness is frequently discussed in computational terms, yet there is little agreement on what computation is assumed to explain. Debates often oscillate between attempts to compute consciousness itself and claims that consciousness resists any computational account.

This paper argues that this impasse arises from a misplaced framing. Rather than asking whether consciousness can be computed, we propose reexamining what computation operates on, from which perspective, and under which conditions it becomes relevant to conscious experience.

We introduce the notion of a *computational constraint space* to characterize a regime in which computation is shaped by memory, prediction, and internally generated constraints. Within such a regime, computation no longer proceeds solely under externally imposed conditions, but begins to treat its own prior operations as conditions for further computation. Consciousness, on this view, is not a computational object or output, but an organizational state of computation itself.

By distinguishing external modeling perspectives, system-internal computational processes, and natural or evolutionary constraints, we clarify why existing theories of consciousness often talk past one another. This framework allows for a reframing of the so-called hard problem as a mismatch of explanatory levels, rather than an empirical gap.

Finally, we briefly discuss implications for artificial intelligence and robotics, arguing that systems need not possess consciousness in order to engage meaningfully with humans, but must be capable of understanding constraint-mediated interaction, including emotionally structured exchanges.

The aim of this work is not to offer a complete theory of consciousness, but to provide a coherent conceptual map for relating computation, experience, and constraint without conflating levels of analysis.

1 Introduction

Consciousness has long resisted systematic explanation. Despite extensive progress in neuroscience, cognitive science, and artificial intelligence, there remains little consensus on what consciousness is, how it relates to computation, or even how the question should be posed.

One reason for this persistent difficulty is that discussions of consciousness often conflate distinct explanatory perspectives. Questions about computation, subjective experience, natural law, and evolution are frequently addressed within a single argumentative space, without clarifying which level of description is operative.

This paper proposes that progress requires a reorientation of the problem. Rather than asking whether consciousness itself can be computed, we ask a prior and more precise set of questions: Who or what treats consciousness as a computational object? What aspects of cognitive organization are subject to computation? And under what conditions does computation become relevant to conscious experience at all?

To navigate these questions, we introduce a guiding distinction between three perspectives: external modeling perspectives, system-internal computational perspectives, and natural or evolutionary perspectives. Much of the disagreement in the literature, we argue, arises from shifting implicitly between these perspectives without acknowledging the shift.

Within this framework, the central thesis of the paper can be stated succinctly: consciousness is not a computational object, nor a state produced by computation. Rather, it is a regime of computation in which computational processes begin to treat their own operations as conditions for further computation. We refer to this regime as a *computational constraint space*.

Developing this claim requires separating what computation operates on from how conscious experience is organized. Accordingly, the paper proceeds by first clarifying how consciousness comes to be treated as a computational object at all, and by whom. We then examine what aspects of cognition are genuinely computationally tractable, and show that these concern conditions, constraints, and structures, rather than subjective experience itself.

The core of the argument is developed through an analysis of memory, prediction, and constraint as the computational substrate from which self-conditioning computation can emerge. Recognition alone is shown to be insufficient. Only when computational processes acquire temporal depth, recursive accessibility, and internally generated constraints does a transition occur toward a regime capable of supporting consciousness.

The paper further situates this transition within the context of natural laws and evolution, arguing that computational constraints are not freely designed, but are embedded through long-term interaction with environmental and bodily regularities. This perspective allows for a reframing of the so-called hard problem of consciousness, which we interpret as a consequence of a mismatch between explanatory levels.

Finally, the implications of this framework for artificial intelligence and robotics are briefly discussed, with particular attention to the distinction between possessing consciousness and understanding emotionally mediated constraints in interaction.

Throughout, the aim is not to offer a complete theory of consciousness, but to provide a coherent map of how computational explanations can be meaningfully related to conscious experience without conflating levels of analysis.

In this sense, the guiding hypothesis of this paper is that consciousness is the state in which computation operates within a self-conditioning constraint space.

2 Background: Consciousness and Computation

The relationship between consciousness and computation has been approached from multiple disciplinary perspectives, often without explicit agreement on what is meant by "computation" itself. As a result, debates surrounding computational accounts of consciousness frequently conflate distinct explanatory levels.

In this section, we survey influential philosophical and cognitive scientific approaches to consciousness with a particular focus on how computation is implicitly or explicitly understood. To organize this survey, we adopt a tripartite distinction that anticipates the analysis developed in later sections: external modeling perspectives, system-internal computational perspectives, and natural or evolutionary perspectives.

2.1 External Modeling Perspectives

Early discussions of machine intelligence, most notably those inspired by formal theories of computation, framed cognition in terms of symbol manipulation. In this tradition, computation was

understood as rule-governed transformation over discrete representations. From an external modeling standpoint, the central question became whether a system implementing such transformations could exhibit behavior indistinguishable from human cognition.

Philosophical critiques of this approach often targeted the gap between formal manipulation and meaning. Arguments emphasizing this gap challenged the idea that computation alone, understood as syntactic processing, could account for conscious understanding. Importantly, however, these critiques typically operated at the level of external description: they questioned whether computational models, as explanatory tools, were sufficient to capture consciousness.

From this perspective, consciousness appears as an object of theoretical modeling, and computation functions as a methodological lens. The computational status of consciousness here is a property of the explanatory framework, not of the system itself.

2.2 System-Internal Computational Perspectives

A different strand of research shifted attention from symbolic manipulation to internal information processing mechanisms. In cognitive science and neuroscience, computation came to be associated with processes such as perception, memory update, decision-making, and motor control.

Within these frameworks, computation is no longer an abstract formal operation, but a physically realized process embedded in neural or embodied dynamics. Consciousness is approached indirectly, through functional roles, global availability of information, or patterns of integration across subsystems.

While these accounts differ substantially in detail, they share a common assumption: that computation operates over internal states that guide behavior and cognition. However, even in these models, consciousness itself is rarely identified as a distinct computational variable. Instead, it is inferred from the organization, accessibility, or integration of computational processes.

2.3 Natural and Evolutionary Perspectives

A third perspective situates computation within the broader context of natural law and evolution. Here, computation is not treated as a design choice or representational strategy, but as a consequence of physical and biological constraints.

From this standpoint, cognitive systems are shaped by the need to operate reliably under energetic, temporal, and environmental limitations. Computational structures emerge through evolutionary processes that favor stability, adaptability, and efficiency.

Consciousness, within this view, is not explicitly selected or implemented. Rather, it arises as an organizational byproduct of increasingly complex regulatory mechanisms. Computation is thus understood as embedded in material processes and constrained by the regularities of the natural world.

2.4 Toward a Clarified Notion of Computation

Across these perspectives, "computation" refers to markedly different phenomena: formal symbol manipulation, neural information processing, and constraint-driven physical dynamics. Failure to distinguish these senses has contributed to persistent confusion in debates about consciousness.

The present work does not privilege any single computational formalism. Instead, it treats computation as a general notion encompassing state transitions governed by constraints, whether symbolic, neural, or physical.

By making explicit which perspective is operative in a given argument, we can avoid misplaced expectations about what computation should explain. This clarification prepares the ground for

the subsequent analysis of who treats consciousness as a computational object and what aspects of cognition are genuinely subject to computation.

2.5 Canonical Theories and Their Computational Commitments

The distinctions introduced above can be made more concrete by briefly situating several influential theories of consciousness with respect to their implicit commitments about computation. The purpose of this mapping is not to evaluate these theories, but to clarify the sense in which each treats computation as explanatorily central.

Global Workspace–type theories typically understand computation in terms of information access and broadcast. Here, computation is associated with the selective amplification and global availability of representations. Consciousness, from this perspective, is linked to the functional role played by information once it becomes accessible to multiple subsystems.

Integrated Information approaches shift the focus from access to structure. Computation is understood as the causal organization of a system, quantified in terms of integration. What matters is not how information is used, but how tightly system components constrain one another. Consciousness is associated with a particular organizational profile of computational interactions.

Predictive Processing and related free-energy–based frameworks interpret computation as the minimization of prediction error under hierarchical generative models. Here, computation is inherently dynamical and temporally extended. Rather than producing discrete outputs, computational processes continuously reshape the space of expected states and actions.

Enactive and embodied approaches often resist the language of computation altogether. Nevertheless, when interpreted operationally, they can be seen as emphasizing constraint-driven sensorimotor dynamics. Computation, in this sense, is not symbol manipulation but the regulation of viable interactions between organism and environment.

These approaches differ substantially in their explanatory targets and formal tools. However, they can be distinguished most clearly by the level at which computation is assumed to operate: as external modeling abstraction, as system-internal information processing, or as an emergent property of embodied interaction constrained by natural law.

The present work does not adopt any of these theories wholesale. Instead, it draws on their shared insight that computation is inseparable from constraint, while reframing consciousness as a regime in which such constraints become recursively self-conditioning.

3 What Treats Consciousness as a Computational Object?

Before asking what aspect of consciousness can be computed, it is necessary to clarify a prior question: what treats consciousness as a computational object in the first place?

Much of the debate surrounding computational theories of consciousness implicitly assumes a single perspective. However, this assumption obscures a critical distinction between different levels at which consciousness may be described, modeled, or enacted.

In this section, we distinguish three perspectives that are often conflated but play fundamentally different roles in the interpretation of consciousness as a computational phenomenon.

3.1 The External Modeling Perspective

From the standpoint of scientists, philosophers, and engineers, consciousness appears as an explanatory target. At this level, consciousness is treated as a phenomenon to be modeled, measured, or functionally approximated.

When consciousness is described as a computational object in this context, the claim is methodological rather than ontological. It reflects a decision to apply computational tools and abstractions to the study of cognitive systems.

This external modeling perspective is indispensable for theory construction. However, it does not imply that the system under study itself represents or computes consciousness as an object.

3.2 The System-Internal Perspective

From the internal perspective of a cognitive system, there is no explicit computational object corresponding to consciousness. The system computes predictions, evaluates constraints, updates memory, and selects actions.

These operations may give rise to conscious experience, but they do not operate over a representation labeled "consciousness." To assume otherwise is to project the external modeling perspective onto the internal dynamics of the system.

This distinction is crucial. Confusing internal computation with external description leads to the mistaken expectation that consciousness must appear as a discrete internal variable or state.

3.3 The Natural and Evolutionary Perspective

A third perspective operates at a deeper level still. From the standpoint of natural laws and evolutionary processes, there is no entity corresponding to consciousness that is treated as a computational object.

Instead, evolutionary dynamics shape which computational architectures are viable. Physical and statistical regularities constrain how information can be processed, stored, and acted upon.

From this perspective, what emerges as consciousness is not targeted or computed, but arises as a byproduct of self-stabilizing computational regimes that prove adaptive over time.

3.4 Resolving the Perspective Shift

These three perspectives serve different explanatory functions and should not be conflated.

When consciousness is treated as a computational object, this treatment occurs only at the level of external modeling. Internally, computation operates on memory, prediction, and constraint. At the natural and evolutionary level, computation itself is shaped by forces indifferent to phenomenology.

Recognizing this shift in perspective clarifies why debates about the computability of consciousness often generate confusion. The disagreement frequently concerns not computation itself, but the level at which computation is assumed to apply.

By making this distinction explicit, we prepare the ground for the subsequent analysis of what computation actually operates on and how a self-conditioning computational regime may emerge without positing consciousness as an internal computational object.

4 What Aspect of Consciousness Is Computed?

Discussions of computational accounts of consciousness often begin by asking whether consciousness itself can be computed. This question, however, already presupposes a misleading object-centered framing.

In the framework developed in this paper, consciousness is not treated as a computational object, state, or output. Accordingly, it is not consciousness as such that is computed. Instead, computation operates on the conditions, structures, and constraints under which conscious experience may arise.

4.1 Computation Targets Conditions, Not Experience

What is computationally tractable are not subjective experiences themselves, but the functional and dynamical conditions that shape cognitive activity. These include, but are not limited to:

- the organization and reuse of memory,
- the propagation of predictive constraints,
- the regulation of action under uncertainty,
- and the stabilization of internal dynamics.

Such processes are observable, implementable, and open to formal analysis. They define how a system transitions between internal states and how those transitions are constrained over time.

By contrast, subjective experience is not an additional variable that computation produces or manipulates. Treating experience as a computational output conflates the operational level of description with the organizational regime in which computation unfolds.

4.2 From State Transitions to Constraint Regimes

Standard computational models describe cognition in terms of state transitions governed by fixed or externally specified rules. Within such models, it is natural to search for a state or representation corresponding to conscious experience.

However, as argued in later sections, the relevant explanatory shift is not from one state to another, but from fixed transition rules to dynamically evolving constraint regimes.

What is computed, in this sense, are not isolated transitions, but patterns of admissible transitions. These patterns determine which states are reachable, which distinctions are salient, and which actions are viable.

4.3 Computational Accessibility and Recursive Conditions

A further distinction must be drawn between what is computed and what is computationally accessible.

Certain internal processes— such as memory reactivation, prediction error modulation, or constraint adjustment— become accessible to the system itself only when computation acquires temporal depth and recursive structure.

This accessibility does not imply explicit self-representation. Rather, it reflects the fact that past computational activity can function as a condition for present and future computation.

In this sense, what computation increasingly operates on are its own historically generated constraints. This shift prepares the ground for understanding consciousness not as a computed entity, but as a regime in which computation becomes conditionally self-referential.

4.4 Positioning the Question

The question, then, is not whether consciousness can be computed, but which aspects of cognitive organization are subject to computation and how their interaction gives rise to a self-conditioning regime.

By clarifying what computation targets, this section establishes the conceptual boundary within which subsequent analyses operate. It sets the stage for examining how memory, prediction, and constraint constitute the computational substrate from which consciousness may emerge.

5 Memory, Prediction, and Constraint as the Computational Substrate

In order to ground the proposed view of consciousness, it is necessary to identify the computational substrate from which such a phenomenon could emerge. Rather than treating consciousness itself as a computational object, we focus on three tightly coupled processes: memory, prediction, and constraint. Together, these processes define the conditions under which computation can become recursively self-conditioning.

5.1 Memory as a Computational Workspace

Memory is often described as a storage mechanism for past information. However, in cognitive and biological systems, memory functions less as passive storage and more as a reusable computational workspace.

Rather than preserving past states verbatim, memory enables the selective reactivation, recombination, and reinterpretation of prior computational structures. This reuse allows the system to operate over extended temporal horizons without requiring explicit representations of time itself.

Crucially, such a workspace is a prerequisite for self-reference. Only when past internal states can be re-accessed as inputs to ongoing computation does it become possible for a system to treat its own prior operations as objects of further processing. In this sense, memory establishes the minimal temporal depth required for recursive cognitive dynamics.

5.2 Prediction as Constraint Propagation

Prediction is commonly understood as an attempt to infer future states of the world. In the present framework, prediction plays a different and more fundamental role.

Rather than forecasting specific outcomes, prediction functions as a mechanism for propagating constraints over possible states and actions. By continuously evaluating discrepancies between expected and actual inputs, the system narrows the space of viable interpretations and behaviors.

Prediction thus mediates between recognition and action. It does not merely describe the world, but actively limits what the system can treat as possible or relevant. Through this limiting function, prediction shapes both perception and decision-making within a unified computational process.

5.3 Constraint as Internalized Natural Regularities

Constraints do not originate within the cognitive system itself. They are imposed by the statistical and physical regularities of the environment, as well as by the system's own embodiment.

Over evolutionary and developmental timescales, these external regularities are compressed and internalized into the system's computational architecture. What appears as an internal constraint is therefore best understood as an embedded trace of natural laws and environmental structure.

Computation, from this perspective, is not an abstract symbol manipulation process, but an operation over internally encoded regularities that reflect the structure of the natural world. The notion of "embedding" is thus justified not metaphorically, but operationally: constraints are literally incorporated into the system's dynamics.

5.4 From Computational Processes to Self-Conditioning

Individually, memory, prediction, and constraint are insufficient to give rise to consciousness. Their significance emerges only when they are tightly integrated.

When memory enables the reuse of prior internal states, prediction propagates constraints over possible futures, and constraints stabilize the overall computational regime, the system begins to operate under conditions that are partially generated by its own activity.

At this point, computation no longer proceeds solely under externally imposed conditions. Instead, the system's own operations become conditions for further computation. It is this transition—from externally conditioned computation to self-conditioning computation—that sets the stage for the emergence of consciousness.

This transition does not introduce a new computational object. Rather, it establishes a new regime of computation, one in which the system's internal dynamics are recursively accessible and modifiable.

6 Is Consciousness a Problem of Recognition?

6.1 Recognition Without Reflexivity

A common assumption in theories of mind is that consciousness can be explained primarily in terms of recognition or perception. On this view, to be conscious is to recognize objects, states, or events in the environment. However, when examined in light of the computational substrate outlined in Section 5, this assumption proves insufficient.

Recognition, by itself, describes only a partial aspect of cognitive processing. A system may successfully classify inputs, map sensory patterns to internal categories, and even guide appropriate actions, without exhibiting anything that warrants the label of consciousness.

6.2 Temporal Depth and Recursive Access

The limitation of recognition-based accounts lies in their neglect of recursive structure. Recognition operates on external inputs, but does not, on its own, provide a mechanism for a system to treat its own internal operations as objects of further computation.

From the perspective developed in this paper, consciousness requires more than accurate or sophisticated recognition. It requires that recognition be embedded within a temporally extended computational process supported by memory, prediction, and constraint.

Memory introduces temporal depth, allowing past internal states to be reactivated and reinterpreted in the present. Prediction propagates constraints across possible future states, binding perception to action within a unified process. Constraints stabilize this dynamic by internalizing environmental and bodily regularities.

6.3 From Recognition to Self-Conditioning

Only when recognition is situated within this triad does a crucial transition occur: the system begins to recognize not only the world, but also the conditions under which it recognizes the world.

At this point, recognition becomes reflexive.

This reflexivity is inseparable from the construction of a self-model. The self, in this sense, is not a pre-given entity, but an emergent structure arising from the system's ability to track, reuse, and modulate its own computational states over time. Temporal continuity and self-reference are therefore not additional features layered onto recognition; they are structural consequences of recursive access to memory and predictive constraints.

Thus, consciousness is not reducible to recognition. Rather, it emerges when recognition is subsumed within a self-conditioning computational regime, where internal processes become conditionally accessible to the system itself.

This reframing clarifies why purely perceptual or representational accounts of consciousness fall short. They address what the system recognizes, but not how the act of recognition becomes part of the system's own operational conditions.

7 Natural Laws, Evolution, and Embedded Computation

The computational processes discussed so far do not arise in isolation. Memory, prediction, and constraint are not arbitrary design choices, nor are they purely internal constructions of a cognitive system. They are shaped, limited, and stabilized by natural laws and by evolutionary processes operating over extended timescales.

Understanding consciousness as a self-conditioning computational regime therefore requires examining the origin of its constraints.

7.1 Natural Laws as External Constraints

At the most fundamental level, all computation performed by a cognitive system is constrained by physical and statistical regularities of the world. These regularities determine which transformations are possible, which signals are reliable, and which actions lead to stable outcomes.

Importantly, natural laws do not prescribe specific representations or algorithms. Instead, they define a space of viable interactions within which any adaptive system must operate. From this perspective, natural laws function as external constraints that shape the landscape of possible computation without being explicitly represented.

7.2 Evolutionary Compression of Regularities

Evolution provides the mechanism by which external constraints become internalized. Through processes of variation, selection, and stabilization, regularities in the environment are gradually compressed into the structure of the organism.

This compression is neither symbolic nor explicit. It manifests as biases in perception, regularities in neural dynamics, and preferred patterns of action. What appears as an internal computational constraint is thus the result of long-term adaptation to stable features of the natural world.

Computation, in biological systems, can therefore be understood as operating over representations and dynamics that already embody environmental structure. This perspective shifts the explanatory burden from representation to constraint.

7.3 Embedding as Operational Internalization

The notion of embedding is often treated metaphorically. Here, it should be understood in a strictly operational sense.

To embed a regularity is not to store a description of it, but to reorganize computational dynamics such that the regularity is implicitly respected. Constraints embedded in this way do not need to be consulted or evaluated; they are enacted by default.

This view clarifies how natural laws can be present within computation without being explicitly modeled. They are embedded as tendencies, limits, and attractors within the system's internal dynamics.

7.4 From Embedded Constraints to Self-Conditioning

Once constraints are embedded, they no longer function solely as external limitations. They become part of the internal conditions under which computation proceeds.

When combined with memory and prediction, these embedded constraints allow the system to regulate its own operations in light of both past states and anticipated futures. At this stage, computation is no longer merely constrained by the world; it is constrained by its own historically shaped dynamics.

This transition marks a critical step toward self-conditioning computation. The system does not represent natural laws, but operates as if they were internal. It is within this regime that consciousness can later be understood as an emergent computational constraint space.

8 Consciousness as a Computational Constraint Space

Having examined the computational substrate (Section 5), the limitations of recognition-based accounts (Section 6), and the role of natural laws and evolution in shaping embedded computation (Section 7), we are now in a position to articulate a unified account of consciousness.

The central claim of this paper is that consciousness is neither a primitive mental substance nor a computational object directly manipulated by the system. Instead, consciousness is best understood as a computational constraint space that emerges when computation becomes self-conditioning.

8.1 Beyond Objects and Algorithms

Traditional debates often assume that consciousness, if computationally explicable, must correspond to a specific data structure, algorithm, or representational content. This assumption is misleading.

In the framework developed here, consciousness does not reside in any particular component. It is not identical to memory, prediction, or constraint, nor to any specific configuration thereof. Rather, it characterizes a regime of computation in which these processes are organized such that internal operations function as conditions for subsequent computation.

Consciousness, therefore, is not computed; it is the mode under which computation proceeds.

8.2 Constraint Space and Self-Conditioning

A constraint space defines the limits, tendencies, and permissible transformations of a computational process. In simple systems, such constraints are imposed externally. In more complex biological systems, constraints are increasingly internalized through memory, predictive structure, and evolutionary embedding.

When a system begins to operate under constraints that are themselves products of its own prior computational activity, a qualitative transition occurs. Computation becomes self-conditioning: its past states, inferred futures, and embedded regularities jointly determine the space of possible present operations.

This self-conditioning does not require explicit self-representation. It arises from recursive accessibility to internal constraints across time. The resulting constraint space is dynamic, historically structured, and partially opaque to the system itself.

8.3 Consciousness as an Emergent Regime

Within this regime, the system's internal dynamics exhibit properties commonly associated with consciousness: temporal continuity, a unified but revisable perspective, and sensitivity to its own operational history.

These properties are not added as separate modules. They emerge from the way computation is constrained. From this perspective, consciousness is neither an illusion nor an inexplicable addition to cognition. It is an emergent organizational state of a self-conditioning computational system.

Importantly, this account does not deny the reality of subjective experience. Rather, it relocates the explanatory task: from identifying a special object of experience to understanding the conditions under which computational processes become recursively entangled with their own constraints.

8.4 Positioning the Account

By framing consciousness as a computational constraint space, this account avoids several long-standing dichotomies: between computation and phenomenology, between functional explanation and subjective experience, and between natural law and mental autonomy.

At the same time, it deliberately leaves open critical questions concerning the precise boundary conditions under which such a regime arises. These open issues are not deficiencies, but indicators of where further theoretical and empirical work must be directed.

9 Reframing the Hard Problem

The so-called "hard problem of consciousness" is often presented as the decisive obstacle for any computational or functional account of mind. It asks why and how physical or informational processes are accompanied by subjective experience.

From the perspective developed in this paper, this problem arises not from an empirical gap, but from a category error in the formulation of the question itself.

9.1 The Category Error Behind the Hard Problem

The hard problem presupposes that subjective experience must be explained at the same descriptive level as neural mechanisms, algorithms, or representations. It therefore demands a causal account linking objective processes directly to phenomenal qualities.

However, as argued in the preceding sections, consciousness is not an object, state, or output of computation. It is a computational constraint space: a regime in which computation becomes self-conditioning through memory, prediction, and embedded constraints.

Demanding a causal explanation of experience at the level of functional components is therefore a mismatch of explanatory levels. Functional constraints do not generate experience as an effect; they define the conditions under which experience can arise as an emergent organizational property.

9.2 Qualia and the Misplaced Demand for Causation

When consciousness is treated as a computed object, it is natural to ask where qualia are located and what causes them. Once consciousness is understood as a constraint space, this demand loses its footing.

Qualia are not entities produced by computation, nor properties added to otherwise complete processes. They reflect the system's operation within a self-conditioning computational regime, characterized by temporal continuity, recursive accessibility, and sensitivity to its own internal constraints.

From this standpoint, the hard problem does not reveal a mysterious residue left unexplained by computation. Rather, it exposes the limitations of an object-centered and causation-driven explanatory framework when applied to emergent computational regimes.

This reframing does not dissolve subjective experience nor deny its reality. Instead, it relocates the explanatory task: from identifying a causal origin of qualia to understanding the structural conditions under which computational systems become experientially organized.

10 Conclusion

This paper has argued for a reframing of consciousness that departs from object-centered and representation-based accounts. Rather than treating consciousness as a primitive phenomenon or as a computational object to be generated or localized, we have proposed understanding it as a computational constraint space emerging when computation becomes self-conditioning.

By analyzing the roles of memory, prediction, and constraint, we showed how computation can acquire temporal depth, recursive accessibility, and sensitivity to its own operational history. Recognition alone is insufficient for this transition. Only when recognition is embedded within a dynamically constrained computational regime does a system begin to operate under conditions partially generated by itself.

From this perspective, natural laws and evolutionary processes do not merely provide background conditions for cognition. They shape and compress regularities that become embedded within computational dynamics, thereby enabling systems to regulate their own operations over time. Consciousness, in this sense, is neither imposed by nature nor freely constructed by cognition, but arises at their intersection.

This framework also allows for a principled reframing of the so-called hard problem of consciousness. Rather than demanding a causal explanation of subjective experience at the level of functional components, we locate the source of the difficulty in a mismatch of explanatory levels. Once consciousness is understood as an emergent computational regime, the expectation of a direct causal account of qualia loses its conceptual necessity.

At the same time, several important challenges remain. Most notably, the present account emphasizes structural and organizational conditions without specifying precise boundary criteria for when self-conditioning computation becomes phenomenally conscious. Determining how degrees of recursive accessibility, temporal integration, and constraint internalization relate to qualitative differences in experience remains an open problem.

These challenges should not be seen as shortcomings, but as indicators of where future work must focus. Further progress will require both formal analyses of self-conditioning computation and empirical investigation into how memory, prediction, and constraint are instantiated in biological and artificial systems.

In conclusion, we propose that consciousness is best understood not as something computation produces, but as the state in which computation begins to treat itself as a condition. This shift in

perspective reorients the study of consciousness from the search for elusive mental objects to the analysis of the conditions under which computation becomes experientially organized.

In this sense, consciousness is not something computation produces, but the state in which computation operates within a self-conditioning constraint space.

A A Minimal Formalization of Computational Constraint Space

This appendix provides a minimal mathematical characterization of what is meant by a *computational constraint space* in the main text. The goal is not to offer a complete formal model of consciousness, but to clarify the sense in which the proposed notion is computational rather than metaphorical.

A.1 Computational Process

Let $\mathcal{S}$ be a state space, and let a computational process be defined as a (possibly stochastic) state transition rule

$$T : \mathcal{S} \times \mathcal{I} \to \mathcal{S},$$

where $\mathcal{I}$ denotes inputs (e.g., sensory signals or internal activations).

In standard computational models, T is fixed, and computation proceeds by iterating transitions under externally given conditions.

A.2 Constraints

We define a constraint as a restriction on admissible state transitions. Formally, let

$$\mathcal{C} \subseteq \mathcal{S} \times \mathcal{S}$$

be a set of permitted transitions.

A constrained computational process is then one in which transitions satisfy

$$(s_t, s_{t+1}) \in \mathcal{C}.$$

Importantly, constraints need not be explicitly represented. They may be implicit in system dynamics, architecture, or parameterization.

A.3 Constraint Space

The *constraint space* $\mathbb{C}$ is defined as the space of all constraints that effectively govern the system's transitions. That is,

$$\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots\},$$

where each $\mathcal{C}_i$ restricts the set of admissible transitions in a distinct manner.

In simple systems, $\mathbb{C}$ is fixed and externally imposed. In adaptive systems, $\mathbb{C}$ may change over time as a result of learning or structural modification.

A.4 Self-Conditioning

A computational process is said to be *self-conditioning* if elements of its constraint space are functions of its own past states or transitions.

Formally, let

$$\mathcal{C}_t = F(\mathcal{H}_t),$$

where $\mathcal{H}_t = \{s_0, s_1, \ldots, s_t\}$ is the system's state history, and F is a constraint-generating functional.

In such systems, the admissible transitions at time t depend on the system's own computational history, rather than solely on external conditions.

A.5 Computational Constraint Space

We define a *computational constraint space* as a computational regime characterized by:

- a state space $\mathcal{S}$,
- a transition rule T,
- a dynamically evolving constraint space $\mathbb{C}$,
- and self-conditioning dynamics such that $\mathcal{C}_t \in \mathbb{C}$ is partially determined by $\mathcal{H}_t$.

This definition does not posit a specific representational format, objective function, or optimization principle. It captures only the minimal structure required for computation to operate under conditions generated by its own activity.

A.6 Relation to the Main Thesis

Under this definition, consciousness is not identified with any particular state $s \in \mathcal{S}$ nor with any specific constraint $\mathcal{C}$. Instead, consciousness corresponds to a regime in which computation unfolds within a self-conditioning constraint space.

This formalization is intentionally minimal. It serves to delimit the conceptual commitments of the main argument, while leaving open how such structures are instantiated in biological or artificial systems.

B Implications for AI and Robotics: Self-Conditioning Computation

This appendix outlines selected implications of the proposed framework for the design and interpretation of artificial intelligence and robotic systems. These implications are not direct claims of the main argument, but follow naturally from understanding cognition and consciousness as regimes of self-conditioning computation.

B.1 Beyond Task-Oriented Computation

Most contemporary AI systems operate under externally specified objectives, fixed architectures, and static constraints. Even adaptive systems typically modify parameters within a computational regime whose governing conditions remain externally imposed.

From the perspective of computational constraint spaces, such systems do not qualify as self-conditioning in the sense discussed in this paper. Their computations adapt, but the conditions of adaptation are not themselves products of the system's own operational history.

This observation does not diminish their utility. Rather, it clarifies the distinction between task-oriented computation and systems capable of regulating their own computational conditions.

B.2 Self-Conditioning as a Design Principle

If self-conditioning computation is taken as a guiding principle, the design focus shifts away from explicit representations of mental states toward mechanisms that allow internal constraints to evolve as a function of the system's own activity.

In artificial systems, this may take the form of:

- memory structures that influence future processing pathways,
- predictive mechanisms that reshape internal action spaces,
- constraint-modulating dynamics that persist across operational timescales.

Such systems need not possess consciousness in any strong phenomenological sense. However, they exhibit a form of internal coherence that more closely resembles biological cognition than purely reactive architectures.

B.3 Understanding Without Feeling: Emotional Exchange

A particularly relevant implication concerns the role of emotion in human–robot interaction.

It is not necessary for a robot to possess emotions in order to function effectively in social or cooperative contexts. What is required, however, is the ability to *understand* emotional expressions as constraints on interaction.

Within the present framework, emotions can be interpreted not as internal states to be replicated, but as externally observable signals that modify the constraint space of ongoing interaction. Human emotional expressions reshape the space of acceptable actions, interpretations, and responses.

A robot capable of self-conditioning computation need not "feel" emotion. It must instead be able to incorporate emotionally mediated constraints into its own operational dynamics. This constitutes a form of emotional understanding without emotional embodiment.

B.4 Emotional Exchange as Constraint Coordination

From this standpoint, emotional exchange can be modeled as a coordination process between constraint spaces. Human agents continuously adjust their expectations and actions based on perceived emotional cues. A robot that can track, interpret, and respond to these adjustments participates in emotional exchange at a functional level.

Importantly, this participation does not require the robot to generate subjective experience. It requires sensitivity to how emotional signals reconfigure the interactional constraint space.

This distinction aligns naturally with the main thesis of this paper: what matters for interaction is not shared phenomenology, but compatibility of constraint regimes.

B.5 Limits and Open Questions

The present framework does not claim that self-conditioning computation is sufficient for artificial consciousness, nor that emotional understanding automatically follows from architectural complexity.

Instead, it provides a principled way to separate questions of consciousness from questions of interactional competence. Robots may remain non-conscious systems while still engaging meaningfully with human agents through constraint-sensitive computation.

Clarifying the minimal mechanisms required for such engagement remains an open research problem, bridging cognitive science, robotics, and the philosophy of mind.

C Philosophical and Cognitive Scientific Debates on Consciousness

This appendix provides a structured survey of major philosophical, cognitive scientific, and artificial intelligence–oriented discussions on consciousness. The purpose is not to offer an exhaustive review, but to situate the present work within the broader landscape of debates, including both computational and non-computational approaches.

Throughout this appendix, references to representative works are provided to support and contextualize the discussion. For ease of navigation, a consolidated list of key references, organized by domain, is presented at the end of this appendix (Appendix C.7).

C.1 Early Philosophical Approaches

Early philosophical treatments of consciousness were not framed in computational terms. Instead, they focused on introspection, phenomenal character, intentionality, and the relation between experience and the world.

Phenomenological traditions emphasized the structure of lived experience, temporal continuity, and the irreducibility of first-person perspective. From this standpoint, consciousness was not treated as an object among others, but as the condition under which objects appear at all (see Appendix C.7, Philosophy).

Analytic philosophy, by contrast, approached consciousness through problems such as mental representation, intentionality, and the mind–body relation. Although computation was not initially central, these discussions established conceptual distinctions that later became crucial for debates about functionalism and physicalism.

C.2 Functionalism and Its Critics

With the rise of cognitive science, functionalism proposed that mental states should be individuated by their causal or functional roles rather than by their physical substrate. This shift made it possible to discuss consciousness in relation to system organization and information processing.

At the same time, functionalism faced sustained criticism. Objections focused on the apparent gap between functional description and subjective experience. Arguments concerning understanding, meaning, or phenomenal character challenged the sufficiency of purely functional or computational accounts.

These critiques did not uniformly reject computation, but they raised doubts about whether computation, understood as formal structure or causal role, could fully capture the conditions of conscious experience (see Appendix C.7, Philosophy).

C.3 Computational and Information-Theoretic Accounts

Computational approaches to consciousness diverged along multiple lines. Some emphasized global access, information broadcast, or workspace architectures, while others focused on integration, causal structure, or complexity measures.

In these frameworks, computation is typically understood as system-internal processing that enables flexible behavior, coordination across subsystems, or efficient inference. Consciousness is inferred from organizational or dynamical properties of such processing, rather than treated as a directly computed object (see Appendix C.7, Cognitive Science).

Despite their differences, these approaches share a commitment to explaining consciousness in terms of organizational features of information processing systems.

C.4 Predictive, Enactive, and Embodied Perspectives

Predictive, enactive, and embodied approaches challenge the view of cognition as passive information processing. They emphasize action, sensorimotor coupling, and constraint-driven dynamics between organism and environment.

From these perspectives, consciousness is not localized within internal representations, but emerges from ongoing interaction shaped by bodily and environmental constraints. Computation, when invoked, is understood broadly as the regulation of viable action under physical and biological regularities (see Appendix C.7, Cognitive Science).

These views often resist strong computational interpretations, yet they align naturally with approaches that emphasize constraint, structure, and condition over symbolic manipulation.

C.5 Non-Computational and Anti-Computational Positions

Some philosophical positions explicitly deny that consciousness can be explained in computational terms. These include various forms of dualism, biological naturalism, and views that treat consciousness as involving irreducible properties.

Such accounts typically argue that subjective experience cannot be captured by functional, informational, or computational description alone. While differing in metaphysical commitment, they highlight a recurring concern: that treating consciousness as a computational object misidentifies the level at which explanation is required (see Appendix C.7, Philosophy).

C.6 Positioning the Present Work

The present work does not align itself exclusively with any single tradition. Instead, it draws on insights from multiple perspectives, while rejecting the assumption that consciousness itself must be the direct target of computation or explanation.

By reframing consciousness as a computational constraint space, this approach aims to clarify why computational, non-computational, and hybrid accounts have often talked past one another. The focus shifts from competing claims about what consciousness is to the conditions under which different explanatory frameworks apply.

C.7 Selected References by Domain

Philosophy of Mind and Phenomenology

- Brentano, F. (1874). *Psychology from an Empirical Standpoint.*

- Husserl, E. (1913). *Ideas Pertaining to a Pure Phenomenology.*
- Merleau-Ponty, M. (1945). *Phenomenology of Perception.*
- Ryle, G. (1949). *The Concept of Mind.*
- Nagel, T. (1974). What is it like to be a bat?
- Dennett, D. (1991). *Consciousness Explained.*
- Chalmers, D. (1996). *The Conscious Mind.*
- Searle, J. (1992). *The Rediscovery of the Mind.*
- Block, N. (1995). On a confusion about a function of consciousness.

Cognitive Science and Neuroscience

- Baars, B. (1988). *A Cognitive Theory of Consciousness.*
- Dehaene, S. (2014). *Consciousness and the Brain.*
- Tononi, G. (2008). Consciousness as integrated information.
- Clark, A. (2013). Whatever next? Predictive brains.
- Friston, K. (2010). The free-energy principle.
- Varela, F., Thompson, E., & Rosch, E. (1991). *The Embodied Mind.*
- Thompson, E. (2007). *Mind in Life.*
- Gallagher, S. (2005). *How the Body Shapes the Mind.*

Artificial Intelligence and Computational Perspectives

- Turing, A. (1950). Computing machinery and intelligence.
- Marr, D. (1982). *Vision.*
- Newell, A., & Simon, H. (1976). Computer science as empirical inquiry.
- Brooks, R. (1991). Intelligence without representation.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach.*
- Bengio, Y. (2017). The consciousness prior.
- Lake, B., Ullman, T., Tenenbaum, J., & Gershman, S. (2017). Building machines that learn and think like people.
- Sutton, R. (2019). The bitter lesson.

D Terminological Clarifications and Conceptual Mapping

This appendix provides terminological clarification and a conceptual mapping of key terms used throughout the paper. Its purpose is to reduce potential confusion arising from overlapping or context-dependent usage, without enforcing artificial uniformity.

D.1 Glossary

Computation State transitions governed by constraints. The term is used broadly to include symbolic, neural, and embodied processes, rather than restricted to formal symbol manipulation.

Constraint A restriction on admissible state transitions, originating from internal structure, learning history, or embedded natural regularities.

Constraint Space The set of constraints that determine which computational transitions are possible. Distinct from any particular state within a state space.

Self-Conditioning A property of computation in which prior computational activity partially determines the conditions under which future computation proceeds.

State A particular configuration of a system at a given time. States are elements of a state space and should not be conflated with constraint spaces.

Regime A mode of operation defined by relatively stable constraint structures. A regime characterizes how computation unfolds over time, rather than a single instantaneous state.

Recognition The classification or identification of inputs or situations. Recognition alone is insufficient for consciousness without recursive and temporally extended structure.

D.2 Conceptual Mapping Across Frameworks

Present Work	Cognitive Science	Philosophy
Constraint Space	Generative Model	Conditions of Experience
Self-Conditioning	Recursive Processing	Self-Reference
Regime	Global State	Mode of Appearance
Computation	Information Processing	Functional Organization
Recognition	Perceptual Categorization	Intentional Act